

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Miha Dodič

**Spletni portal PGD Brezovica pri
Borovnici**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Aleš Smrdel

Ljubljana, februar 2016

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko Univerze v Ljubljani ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko Univerze v Ljubljani izdaja naslednjo nalogo:

Tematika naloge:

Prostovoljna gasilska društva imajo v svoje vrste vključenih veliko članov, ki sodelujejo pri intervencijah, izobraževanjih ter drugih aktivnostih. O svojih aktivnostih morajo gasilska društva voditi evidence in jih posredovati v centralni informacijski sistem. Pogosto pa se zgodi, da so evidence vodene v papirnati obliki. Kot pomoč pri vodenju evidenc v diplomski nalogi razvijte sistem za elektronsko vodenje relevantnih evidenc v okviru nekega gasilskega društva, kot so na primer evidence o opremi, o članih in o intervencijah, ki bo obenem omogočal tudi implementacijo predstavitvene strani društva. Pri implementaciji sistema uporabite arhitekturo model-pogled-nadzornik. Zamislite si podatkovni model in ga implementirajte v relacijski podatkovni bazi, implementirajte nadzorni del ter definirajte in realizirajte vse relevantne prikaze. Pri implementaciji uporabite relevantne tehnologije na strani odjemalce ter strežniško tehnologijo, ki omogoča implementacijo arhitekture model-pogled-nadzornik. Zaradi boljše in takojšnje dostopnosti in možnosti vnašanja podatkov implementirajte tudi odzivni dizajn. Aplikacijo dajte v testiranje tudi uporabnikom in predstavite njihove izkušnje.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Miha Dodič, z vpisno številko **63070070**, sem avtor diplomskega dela z naslovom:

Spletni portal PGD Brezovica pri Borovnici

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Aleša Smrdela,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 3. februarja 2016

Podpis avtorja:

Rad bi se zahvalil vsem, ki so mi v času mojega študija stali ob strani ter me podpirali.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Gasilska organizacija	1
1.2	Informacijska sistema Vulkan in Spin ter vodenje evidenc v PGD	2
1.3	Cilji	6
2	Uporabljene tehnologije in orodja	7
2.1	Microsoft Visual Studio 2015	7
2.2	Arhitektura MVC	9
2.3	Spletne tehnologije na strani odjemalca	11
2.4	Sybase Power Designer	13
3	Podatkovni model	15
3.1	Struktura podatkovnega modela	17
4	Komponente spletnega portala	23
4.1	Spletni vmesnik	23
4.2	Registracija in prijava	26
4.3	Predstavitvena spletna stran	29
4.4	Del za registrirane uporabnike	32

5 Sklepne ugotovitve	41
5.1 Odziv testnih uporabnikov portala	42
5.2 Nadaljnje delo	42
Literatura	45

Seznam uporabljenih kratic

kratica	angleško	slovensko
GZS	Firefighting Association of Slovenia	Gasilska zveza Slovenije
PGD	Voluntary fire brigade	Prostovoljno gasilsko društvo
MVC	Model - View - Controller	Model - pogled - nadzornik
CSS	Cascading Style Sheets	Kaskadne stilske predloge
WYSIWYG	What You See Is What You Get	Kar vidiš, boš tudi dobil
XML	Extensible Markup Language	Razširljiv označevalni jezik
XSLT	Extensible Stylesheet Language Transformations	Jezik razširljivih slogovnih transformacij
HTML	HyperText Markup Language	Jezik za označevanje nadbesedila
XHTML	Extensible HyperText Markup Language	Razširljiv jezik za označevanje nadbesedila
SQL	Structured Query Language	Strukturirani povpraševalni jezik

Povzetek

Naslov: Spletni portal PGD Brezovica pri Borovnici

Diplomsko delo opisuje izdelavo spletnega portala za prostovoljno gasilsko društvo Brezovica pri Borovnici. Ker živimo v dobi interneta in spletnih strani, se je pojavila ideja po izdelavi predstavitvene spletne strani društva. Odločil sem se to idejo nadgraditi in izdelati portal za vodenje vseh potrebnih evidenc, ki so se v preteklosti vodile ročno v papirani obliki. Nekatere evidence se izpolnjujejo po končanih dogodkih v prostorih društva, zato velikokrat pride do vnosa nepopolnih podatkov. V izogib temu je spletni portal prilagojen tudi za delo na mobilnih napravah in tako omogoča vnašanje podatkov med samim trajanjem intervencije. S prilagoditvijo za mobilne naprave pa je pravtako možno do podatkov dostopati kjer koli in kadarkoli. Glavna nit diplomske naloge je izdelava spletnega portala za gasilsko društvo. V prvem delu predstavim uporabljene tehnologije in se ukvarjam z izdelavo primernega podatkovnega modela za hranjenje in povezavo vseh podatkov v spletni aplikaciji. V drugem delu se posvetim sami izdelavi in predstavitvi spletnega portala. Pri izdelavi sem na strani odjemalca uporabil tehnologije HTML, CSS in JavaScript, na strani strežnika pa C# in SQL. Diplomsko zaključim z diskusijo in idejami za nadaljnji razvoj.

Ključne besede: gasilsko društvo, MVC, spletni portal, odzivni dizajn.

Abstract

Title: Portal PGD Brezovica pri Borovnici

The thesis describes the development of a web portal for a voluntary association, namely the fire brigade Brezovica near Borovnica. Since we live in the era of the Internet and websites, the idea for setting up the promotional website of the fire brigade emerged. I decided to upgrade this idea and develop a portal for the management of all necessary records, which were organized manually in the past. Since some of these records are registred after the completion of events, this often lead to incomplete data entry. Therefore, the portal has also been adapted for acces using mobile devices, and enables real-time data entry into the records during the execution of the intervention. With the adjustments enabling easy use on the mobile devices it is also possible to access data anywhere and anytime. Main purpouse of this thesis is to create a web portal for the fire brigade. The first part deals with the production of appropriate data model to store and link all the data into a web application. In the second part I devote the attention to the development and presentation of the web portal. During the development I utilised the HTML, CSS and JavaScript on the client side, while on the server side I used C# and SQL. I conclude the thesis with discussion and ideas for the future work.

Keywords: fire brigade, MVC, online portal, responsive design.

Poglavje 1

Uvod

Ker živimo v dobi interneta in spletnih strani ter želimo imeti na voljo vedno več informacij v čim krajšem času, se je izkazala potreba po izdelavi spletne strani gasilskega društva Brezovica pri Borovnici, katerega član sem tudi sam. Trenutno se vse evidence v društvu vodijo ročno in je dostop do podatkov možen le v gasilskem domu. Evidence nekaterih podatkov pa se interno sploh ne vodijo oziroma jih nekateri člani društva vodijo individualno. Pogosto so se morali člani samo zaradi pridobitve različnih podatkov oglasiti v gasilskem društvu. Prišlo je do ideje, da bi bili ti podatki dostopni preko interneta in tako sem pričel z izdelavo spletnega portala v okviru mojega diplomskega dela.

1.1 Gasilska organizacija

Gasilstvo ima v Sloveniji že 140-letno tradicijo. Imamo edinstven primer prostovoljstva, saj v Gasilski zvezi Slovenije deluje 1348 društev. Ta društva se na nivoju občine ali več občin povezujejo v gasilske zveze, te pa povezuje Gasilska zveza Slovenije (GZS). V GZS je vključenih okrog 150.000 članov[1]. Izmed teh je približno 60.000 operativnih gasilcev, ki posredujejo ob požarih, prometnih nesrečah, naravnih in drugih nesrečah. Preostali del članov predstavljajo: pionirji, mladinci, neoperativni člani, veterani in podporni člani.

Gasilci v zadnjih letih posredujejo vedno manj pri požarnih intervencijah. Pojavljajo pa se nove vrste intervencij, za katere je potrebno pridobiti nova znanja in nabaviti novo opremo. Vso to opremo je potrebno vzdrževati in voditi evidence tako o sami opremi kot tudi o servisnih pregledih. Gasilska zveza Slovenije skrbi za usposabljanje gasilcev v izobraževalnih centrih na Igu, v Sežani in Pekrah. Gasilci tako poleg tečajev za napredovanje v čine opravljajo tudi izobraževanja za posamezna strokovna znanja. V okviru izobraževanja imajo na voljo tečaje za nosilca dihalnega aparata, usposabljanje za gašenje notranjih požarov, tečaj za tehnično reševanje, tečaj za delo z motorno žago in mnoge druge. Zaradi vse večjega porasta drugih nepožarnih posredovanj gasilci opravljajo številna usposabljanja tudi na nivoju gasilskih društev in gasilskih zvez. Tako lahko ustrezno izobraženi gasilci posredujejo pri različnih intervencijah. Krovni organizaciji je potrebno ob vsaki intervenciji posredovati različne podatke, kot so na primer: prisotnost članov na intervenciji, prevoženi kilometri, uporabljena oprema in drugo. Za oddajo teh podatkov se uporablja informacijski sistem Vulkan[2], ki pa je kompleksen za uporabo in okoren za vodenje določenih evidenc (na primer: vodenje evidence prisotnosti članov na društveni vaji ne omogoča takojšnjega vnosa v evidenco prisotnosti na intervenciji). Zaradi tega gasilska društva večino evidenc vodijo ročno in jih hranijo v gasilskih domovih. Ko je kasneje potreben vnos podatkov v informacijski sistem, pa jih iz ročno vodenih evidenc vnesejo v informacijski sistem.

1.2 Informacijska sistema Vulkan in Spin ter vodenje evidenc v PGD

Vsa društva, včlanjena v GZS, imajo dostop do informacijskega sistema Vulkan, v katerem lahko gasilska društva preko za to namenjene spletne aplikacije vodijo svoje evidence. GZS ima centraliziran dostop do evidenc usposobljenosti in opremljenosti gasilskih enot, o usposabljanju članov in drugo. S pomočjo spletne aplikacije se vodi evidence o:

- članih (osebni podatki, splošna in gasilska izobrazba, zadolžitve opreme, zdravstveni pregledi,...),
- opremi (osnovni podatki, lokacija, amortizacije, vloga za sofinanciranje opreme,...),
- organizaciji,
- napredovanjih,
- odlikovanjih za člane in organizacijo,
- usposabljanih in
- intervencijah.

Drugi sistem, ki ga društva uporabljajo, je informacijski sistem Spin[3]. Ta sistem uporabljajo Uprava RS za zaščito in reševanje, regijski centri za obveščanje ter gasilske in druge enote v sistemu zaščite in reševanja. Sistem se uporablja za upravljanje poročil o intervencijah gasilskih enot. Operativec v centru za obveščanje že ob pozivu gasilske enote ustvari dogodek v sistemu, ki ga vodja intervencije zaključi v roku treh dni po intervenciji. Ko je dogodek zaključen v sistemu Spin, se ta avtomatsko prenese še v sistem Vulkan. Sistema sta med seboj le delno povezana. Podatki se prenašajo le iz sistema Spin v sistem Vulkan. Vodja intervencije ob dogodku v sistem Spin vpiše podatek o številu članov na intervenciji in potek dela na intervenciji. Ko je intervencija prenešana v sistem Vulkan, pa lahko šele vpiše poimenski seznam članov, ki so bili prisotni.

V društvu se zaradi potreb po vodenju evidenc dela članov, udeležbe na vajah in intervencijah ter pri pregledih opreme vodijo še druge evidence. Vse te evidence se ponavadi vodijo v papirnati obliki in je tako dostop do teh podatkov omejen. Slika 1.1 prikazuje obrazec za vodenje podatkov o opravljeni intervenciji. Pri izpolnjevanju obrazcev občasno pride tudi do vnosa nepopolnih podatkov. Kot primer lahko navedemo dlje časa trajajočo intervencijo. Ker se evidenca o prisotnosti članov izpolnjuje ob vrnitvi v gasilski

dom, lahko pride do nepopolnega vpisa prisotnih članov. Sistem Vulkan pa ne omogoča takojšnjega vnosa članov, ki so bili prisotni na intervenciji, saj se intervencija vnese v ta sistem šele ko je zaključena v sistemu Spin.

PGD BREZOVICA PRI BOROVNICI

POROČILO O INTERVENCIJI 201_ ŠT. ____

DATUM: _____

VRSTA INTERVENCIJE				
--------------------	--	--	--	--

DAN V TEDNU	ČAS OBVESTILA	ČAS IZVOZA	ČAS PRIHODA NA KRAJ	ČAS ZAKLJUČKA

LOKACIJA INTERVENCIJE	KRAJ	OBČINA
OBSEG POZARA:		

POSKODOVANA ALI UNICENA OPREMA	POSKODOVANI	MRTVI

PORABLJENA GASILNA SREDSTVA				
VOZILA PGD	GVV-1	GVM-1	PMB	DRUGA OPREMA (naprtnjača, gasilniki,...)

PRISOTNE SLUŽBE	ŠT. LJUDI	ŠT. VOZIL	DODATNO OBVEŠČENE DRUGE SLUŽBE:
PGD B p B			
Policija			
Reševalci			

OPIS INTERVENCIJE:			

SODELOVALI NA INTERVENCIJI:			V GASILSKEM DOMU	
1	9	17	1	
2	10	18	2	
3	11	19	3	
4	12	20	4	
5	13	21	5	
6	14	22	6	
7	15	23	7	
8	16	24	8	

POROČILO IZPOLNIL		PODPIS VODJE	
-------------------	--	--------------	--

Slika 1.1: Obrazec za vodenje podatkov o intervenciji.

1.3 Cilji

Zaradi okornosti obeh obstoječih informacijskih sistemov GZS ter želje po dostopnosti društvenih evidenc od kjer koli in kadarkoli, sem se odločil za izdelavo lastnega informacijskega sistema, ki bo to omogočal. Cilji, ki sem jih želel doseči pri izdelavi tega sistema so:

- pokritje vseh osnovnih evidenc, ki so se do sedaj vodile ročno,
- obveščanje zadolženih članov o prihajajočih dogodkih (servis opreme/vozila, potek zdravniškega pregleda,...),
- dosegljivost sistema, kjerkoli in kadarkoli.

Zaradi zgornjih zahtev sem se odločil izdelati spletni portal, ki bo hkrati tudi predstavitevna spletna stran društva.

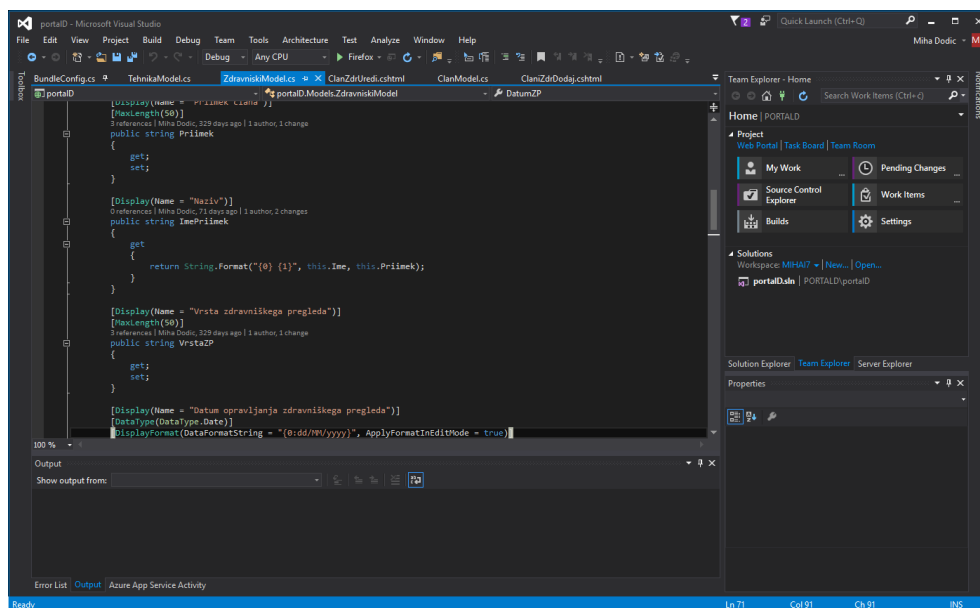
Poglavje 2

Uporabljene tehnologije in orodja

2.1 Microsoft Visual Studio 2015

Visual Studio[4] je integrirano razvojno okolje za razvoj spletnih strani, spletnih aplikacij in namiznih aplikacij. Microsoft je aprila 1995 predstavil prvo različico tega okolja - Visual Studio 4.0. Razvoj okolja aktivno poteka še danes, saj je 30. novembra 2015 izšel prvi popravek za aktualno različico - Visual Studio 2015. Slika 2.1 prikazuje Visual Studio Enterprise 2015. V vsaki različici so bile dodane dodatne funkcionalnosti in podpora novim tehnologijam in programskim jezikom. Visual Studio vsebuje urejevalnik izvorne kode za različne programske jezike. Osnovno so podprti: C, C++, VB.NET, C# in F#. Podporo za druge programske jezike, kot so M, Python, Ruby in drugi, se lahko namesti naknadno. Poleg vsega vsebuje tudi podporo za XML/XSLT, HTML/XHTML, JavaScript in CSS. Za nadzor verzij se lahko uporablja tako Git kot Team Foundation Version Control, saj sta obe tehnologiji popolnoma podprti. V aplikaciji lahko uporabimo različne predloge, pri katerih so določene funkcionalnosti že implementirane. Za potrebe diplomskega dela sem uporabil aplikacijo ASP.NET MVC, v kateri so bila implementirana orodja za delo s člani portala. Funkcionalnosti sem kasneje

prilagodil glede na moje potrebe. Svoje projekte lahko enostavno nadgradujemo preko protokola NuGet[5]. S pomočjo teh paketov v svoj projekt vključimo vse potrebne knjižnice in datoteke.



Slika 2.1: Microsoft Visual Studio Enterprise 2015

2.1.1 ASP.NET

ASP.NET je tehnologija za razvoj spletnih aplikacij z uporabo tehnologij HTML, CSS, JavaScript in izvajanjem skript na strani strežnika. Tehnologija podpira tri različne modele razvoja: *Web Pages*, *MVC* in *Web Forms*. Microsoft je to tehnologijo prvič predstavil januarja 2002 v paketu .NET Framework. Ker se pred izvajanjem vska aplikacija ne glede na jezik, v katerem je bila napisana, prevede v vmesni jezik (*Intermediate Language*), je možno pri pisanju aplikacij ASP.NET uporabljati vse programske jezike iz zbirke .NET (C#, C++, F#, Visual Basic,...)[6].

2.1.2 Nadzor verzij

Za potrebe nadzora verzij pri razvoju spletnega portala sem uporabil tehnologijo Team Foundation Version Control[7]. Za uporabo orodja za nadzor verzij sem se odločil predvsem zaradi dela na dveh različnih delovnih postajah. Z uporabo tega orodja sem imel na obeh napravah popolnoma aktualno različico aplikacije, prav tako pa mi je omogočilo povrnitev na starejše stanje, na primer v stanje, ko je bila aplikacija še delujoča. Team Foundation Version Control podpira porazdeljeno programiranje večjemu številu programerjev na enem ali večih projektih. Omogoča vodenje in končno združevanje posameznih funkcij, datotek ali celotnih projektov.

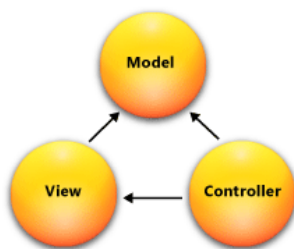
2.1.3 Programski jezik C#

C#[8] je objektno usmerjen programski jezik, ki je zelo razširjen in priljubljen programski jezik na platformi .NET. Jezik je določen s standardi *ECMA-334* in *ISO/IEC 23270:2006* in stremi k enostavnosti, modernosti in splošni uporabi. Zaradi razširjenosti uporabe programskega jezika ima le-ta kvalitetno in obsežno dokumentacijo.

2.2 Arhitektura MVC

Arhitektura MVC (*Model-View-Controller*)[9] razdeli aplikacijo v tri glavne sklope: model, pogled in nadzornik (slika 2.2). V ASP.NET MVC so integrirane vse funkcionalnosti ASP.NET, kot so glavne strani in avtentifikacija glede na članstvo v skupinah. Ta arhitektura nam omogoča, da ločimo različne vidike aplikacije (nadzorna logika, poslovna logika in logika uporabniškega vmesnika), hkrati pa nam omogoča trdno povezavo med temi elementi. Ta vzorec definira, kje naj se nahaja posamezna logika v aplikaciji. Logika uporabniškega vmesnika se nahaja v pogledu (*View*), poslovna logika spada v model (*Model*), nadzorna logika v nadzornik (*Controller*). To ločevanje nam olajša izdelavo aplikacije, saj se lahko posvetimo implemen-

taciji vsakega dela ločeno. Tako se lahko na primer posvetimo izdelavi uporabniškega vmesnika in nismo odvisni od poslovne logike. Neodvisnost logik nam tudi omogoča simultano razvijanje z več razvijalci. Tako lahko en razvijalec dela na pogledu, drugi na vnosni logiki, tretji pa na poslovni logiki.



Slika 2.2: Zasnova MVC arhitekture[9].

2.2.1 Model

Modeli so komponente aplikacije, ki so zadolžene za vzdrževanje ažurnega stanja. To stanje je običajno zapisano v podatkovni bazi. Prav tako skrbi za zapisovanje posredovanih podatkov iz pogleda v podatkovno bazo.

2.2.2 Pogled

Pogledi so komponente, ki prikazujejo uporabniški vmesnik aplikacije. Običajno je uporabniški vmesnik ustvarjen iz podatkov iz modela. Primer je stran urejanja podatkov člana, kjer so besedilna polja, stikala in drugi elementi, napolnjeni s podatki glede na stanje v podatkovni bazi za izbranega člana.

2.2.3 Nadzornik

Nadzorniki skrbijo za upravljanje uporabnikovih interakcij, komunicirajo z modelom in izberejo pogled, ki ga prikaže uporabniški vmesnik. V aplikacijah

MVC pogled namreč samo prikazuje informacije, nadzornik pa se odziva na uporabnikove vnose in interakcije.

2.3 Spletne tehnologije na strani odjemalca

2.3.1 HTML5

HTML[10] je označevalni jezik, ki definira strukturo in semantiko spletnih strani. Trenutno uporabljamo verzijo HTML5, katere končna verzija je bila predstavljena oktobra 2014. V tej različici je dodanih predvsem veliko novosti glede večpredstavnosti, novi tipi vnosnih polj in dvodimenzionalne grafike. Značke HTML so v glavnem sestavljene iz parov (npr. `<a>` in ``), nekatere značke pa so prazne, torej nimajo vsebine in kot take ne nastopajo v paru.

2.3.2 CSS

CSS (*Cascading Style Sheets*)[11] so kaskadne slogovne predloge, ki skrbijo za oblikovanje spletnih strani. Predloge CSS se lahko pojavljajo v treh kaskadah, kot samostojne datoteke, del glave datoteke HTML ali pa so pridružene posameznemu elementu. Bistvo uporabe tehnologije CSS je ločitev strukture strani od njene predstavitve. Tako lažje urejamo sloge, ki se ob uporabi zunanjih slogovnih predlog spremenijo na vseh straneh, ki za oblikovanje uporabljajo to datoteko CSS. S slogovnimi predlogami določamo barve, pisave, odmike, pozicijo in mnoge druge lastnosti. Poleg tega lahko nadziramo tudi aktivnosti nad temi elementi (npr. prekritje gumba z miško).

2.3.3 JavaScript in jQuery

JavaScript[12] je objektni skriptni jezik in nam omogoča ustvarjanje interaktivnih spletnih strani. JavaScript se vključi v HTML kodo in opravlja naloge, ki v samem jeziku HTML niso mogoče. S pomočjo JavaScripta prepoznavamo brskalnike, programiramo obrazce in različne interaktivnosti ter

še veliko več. Uporablja se predvsem v kombinaciji s knjižnico jQuery, ki nam zelo olajša programiranje. JQuery poenostavi kodo, ki je v JavaScriptu dolga in tako omogoča enostavno delo z elementi HTML in CSS, animacijo in odziv na delo uporabnika. JavaScript se izvaja v uporabnikovem brskalniku, vendar različni brskalniki podpirajo različne JavaScript objekte. Za podporo različnih brskalnikov moramo tako napisati več različic naših funkcij.

2.3.4 Bootstrap

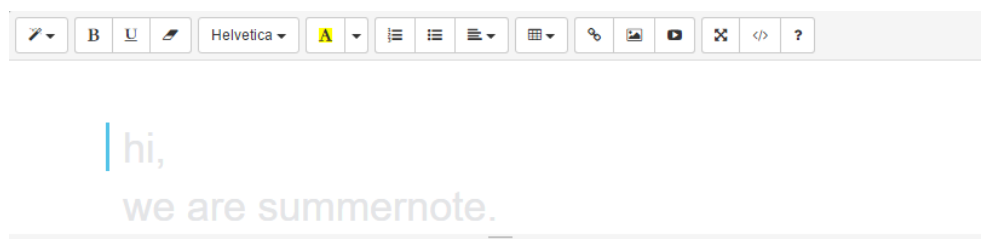
Bootstrap[13] predstavlja odprtokoden skup orodij za kreiranje odzivnih spletnih strani in aplikacij. Temelji na predlogah za tipografijo HTML in CSS, formah, gumbih in navigaciji. Poleg tega vsebuje tudi razširitve v JavaScriptu. Vsi ti sklopi se prilagajajo različnim velikostim brskalnikov, prilagojeni pa so tudi za uporabo na mobilnih napravah, torej na napravah, ki imajo ponavadi zaslone manjših resolucij. Zaradi že oblikovanih prikaznih objektov in form nam omogoča hiter in relativno enostaven razvoj spletnih strani in aplikacij. Bootstrap je bil razvit pri podjetju Twitter v sredini leta 2010. Prvotno je bilo ogrodje razvito za interno rabo v podjetju, kasneje pa se je izkazalo, da so razvili več kot le to. Prva različica ogrodja Bootstrap je bila kot odprtokodno ogrodje objavljena za javnost 19. avgusta 2011. Razvoj ogrodja se še vedno nadaljuje, trenutno pa je v nastajanju različica Bootstrap 4.

2.3.5 Galerija

Za potrebe izdelave interaktivne galerije na predstavitveni strani društva je bila uporabljena prettyPhoto[14] galerija. Gre za galerijo jQuery katera poleg prikaza slik omogoča tudi prikaz videov, flash vsebine in vsebin s portala Youtube. Tako vtičnik prettyPhoto seznam slik spremeni v interaktivno galerijo za prikaz slikovnih vsebin.

2.3.6 Vnosno polje z urejanjem

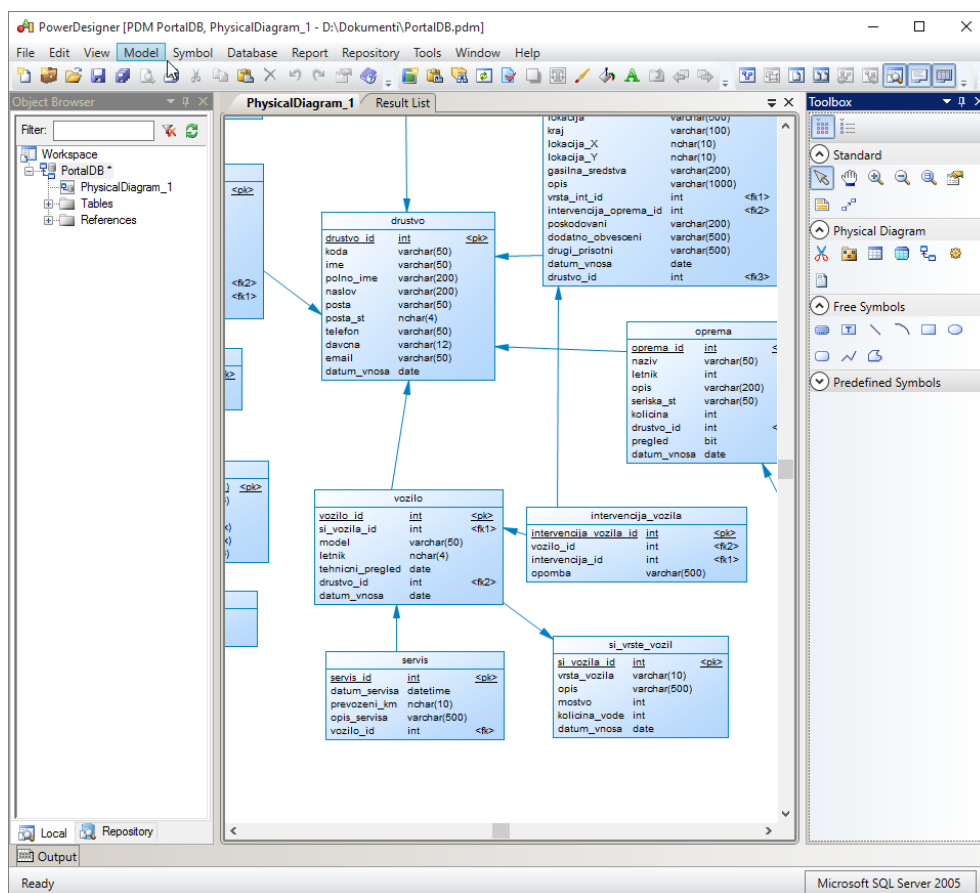
Za potrebe vnosa novic na predstavitveni spletni strani je bil implementiran urejevalnik besedila WYSIWYG. Uporabljen je bil Summernote editor[15], ki je kombinacija jQuery in vtičnika HTML5 za Bootstrap. Z vtičnikom spremenimo klasično besedilno vnosno polje v polje z možnostjo urejanja oblike besedila. Tako dodamo možnosti za spreminjanje velikosti, barve in oblike besedila, postavitev besedila, vstavljanje tabel in slik ter druge funkcije. Vtičnik je popolnoma prilagodljiv, tako da lahko prilagodimo, katere izmed možnosti bodo v urejevalniku omogočene. Vtičnik v ozadju oblikovano besedilo pretvori v besedilo z vsebino HTML. Primer tega urejevalnika je prikazan na sliki 2.3.



Slika 2.3: Primer Summernote urejevalnika besedila.

2.4 Sybase Power Designer

Sybase Power Designer[16] je grafično orodje za izdelavo podatkovnih in drugih modelov. Orodje deluje v operacijskem sistemu Windows, razvilo pa ga je podjetje Sybase, ki je v lasti nemškega podjetja SAP. Power Designer zagotavlja enotno okolje za modeliranje poslovnih procesov, konceptualnih modelov, fizičnih modelov, objektno usmerjenih modelov in modelov XML. V diplomski nalogi je bil Power Designer uporabljen za izdelavo podatkovnega modela in izdelavo diagramov primerov uporabe. Slika 2.4 prikazuje orodje Sybase Power Designer s prikazanim delom razvitega podatkovnega modela.

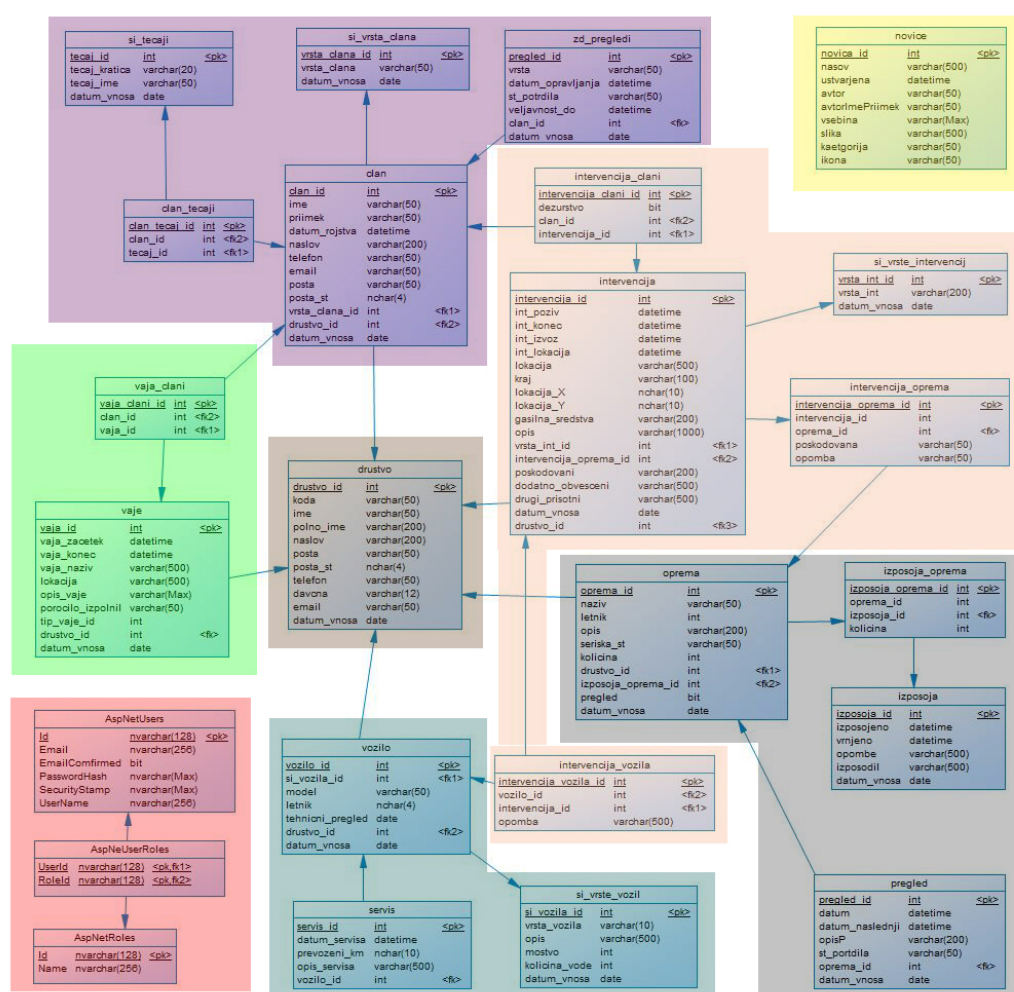


Slika 2.4: Zaslonska slika PowerDesigner-ja 16.5 SP04.

Poglavje 3

Podatkovni model

Razvoj aplikacije sem pričel z zbiranjem podatkov, ki smo jih hoteli voditi v aplikaciji. Ko so bili izbrani vsi okvirni moduli in podatki, ki se bodo hranili v podatkovni bazi, sem pričel z izdelavo entitetno-relacijskega modela podatkovne baze. Podatkovni model sem razvil s programom Sybase Power Designer. Slika 3.1 prikazuje shemo entitetno-relacijskega modela. Podatkovni model je zgrajen tako, da v nadgradnji tukaj predstavljenega sistema omogoča uporabo spletnega portala za več gasilskih društev.



Slika 3.1: Podatkovni model.

3.1 Struktura podatkovnega modela

3.1.1 Društvo

Entiteta **društvo** je osnovna entiteta in vsebuje vse podatke o gasilskem društvu. Atribut *društvo_id* je primarni ključ entitete. Entiteta vsebuje še naslednje attribute: kratko ime gasilskega društva (*ime*), polno ime društva (*polno_ime*), naslov (*naslov*), naziv pošte (*posta*), poštno številko (*posta_st*), telefonsko številko društva (*tels_st*), naslov elektronske pošte (*email*) in davčno številko društva (*davcna*). Poleg tega vsebuje še atribut *koda*, ki bo služil za identifikacijo društva in prikaz pravih podatkov ob implementaciji aplikacije za uporabo v več društvih.

3.1.2 Člani

Sklop vsebuje pet entitet:

- Entiteta **clan** hrani vse glavne podatke o članu društva. Primarni ključ je atribut *clan_id*, ki ga SQL dodeli vsakemu novemu vnosu. Ostali atributi entitete so: ime (*ime*), priimek (*priimek*), rojstni datum (*datum_rojstva*), naslov (*naslov*), telefonska številka (*telefon*), naslov elektronske pošte (*email*), naziv pošte (*posta*), poštna številka (*posta_st*). Atributa *vrsta_clana_id* in *društvo_id* pa članu dodeljujeta vrsto člana in pripadnost društvu iz entitet **si_vrsta_clana** in **društvo**.
- Entiteta **si_vrsta_clana** je šifrant, v katerem so shranjene vse možne vrste članov. Atributi entitete so primarni ključ *vrsta_clana_id*, ki je avtomatsko dodeljen vsakemu vnosu ter atribut opis vrste člana (*vrsta_clana*).
- Entiteta **si_tecaji** je šifrant, v katerem so shranjene vse vrste tečajev in usposabljanj za člane. Atributi entitete so kratica tečaja (*tecaj_kratika*) in polno ime tečaja oziroma usposabljanja (*vrsta_clana*). Atribut *vr-*

sta_clana_id je unikatni identifikator vrste tečaja in je dodeljen samodejno vsakemu vnosu.

- Entiteta **clan_tecaji** je namenjena dodelitvi tečajev (entiteta *si_tecaji*) posameznemu članu (entitea *clan*). Vsak član ima lahko opravljenih več tečajev. Entiteta ima poleg identifikacijskih atributov tečaja (*tecaj_id*) in člana (*clan_id*) še primarni ključ (*clan_tecaji_id*).
- Entiteta **zd_pregledi** hrani podatke o opravljenih zdravniških pregledih članov. Atributi entitete so vrsta zdravniškega pregleda (*vrsta*), datum opravljanja zdravniškega pregleda (*datum_opravljanja*), datum poteka veljavnosti zdravniškega pregleda (*veljavnost_do*), številka potrdila zdravniškega pregleda (*st_potrdila*) in identifikator člana (*clan_id*). Atribut *pregled_id* je unikatni identifikator posameznega zdravniškega pregleda in je avtomatsko dodan vsakemu vnosu.

3.1.3 Vozila

Sklop vsebuje tri entitete:

- Entiteta **vozilo** hrani vse podatke o vozilih društva. Atributi entitete so: model vozila (*model*), letnik proizvodnje (*letnik*), datum tehničnega pregleda (*tehnicni_pregled*), identifikator vrste vozila (*si_vozila_id*) in identifikator pripadnosti društvu (*drustvo_id*). Atribut *tehnicni_pregled* se po poteku avtomatsko podaljša za eno leto. Entiteta vsebuje še primarni ključ vozila (*vozila_id*), ki je avtomatsko dodeljen ob vsakem novem vnosu.
- Entiteta **si_vrste_vozil** je šifrant, v katerem so shranjene vse vrste vozil. Atributi entitete so primarni ključ *si_vozila_id*, ki je avtomatsko dodeljen vsakemu vnosu ter atributi za tip vozila glede na tipizacijo GZS (*vrsta_vozila*), opis vrste vozila (*opis*), moštvo vozila (*mostvo*) in količino vode v vozilu (*kolicina_vode*).

- Entiteta **servis** vsebuje podatke o servisnih pregledih vozil. Atributi entitete so: datum opravljanja servisnega pregleda (*datum_servisa*), število prevoženih kilometrov ob servisnem pregledu (*prevozeni_km*) in opis servisnega pregleda (*opis_servisa*). Entiteta vsebuje še primarni ključ *servis_id*, ki je avtomatsko dodeljen ob vsakem vnosu servisnega pregleda.

3.1.4 Oprema

Sklop vsebuje štiri entitete:

- Entiteta **oprema** vsebuje vse glavne podatke o opremi društva. Atributi entitete so: naziv opreme (*naziv*), leto proizvodnje (*letnik*), podrobnejši opis (*opis*), opcijski atribut serijska številka (*serijska_st*), število kosov opreme (*kolicina*) in atribut, ki nam pove, ali je potrebno za ta kos opreme opravljati periodni servis (*pregled*). Atribut *oprema_id* je primarni ključ entitete in je avtomatsko dodeljen vsakemu vnosu opreme. Atribut *drustvo_id* pa opremi dodeljuje pripadnost društvu iz entitete **drustvo**.
- Entiteta **pregled** vsebuje vse podatke o opravljenih pregledih oziroma servisih opreme. Atributi entitete so: datum opravljenega pregleda (*datum*), datum naslednjega pregleda (*datum_naslednji*), opis pregleda oziroma servisa (*opisP*) in številka potrdila o opravljenem pregledu (*st_potrdila*). Entiteta vsebuje še identifikator društva (*drustvo_id*), ki dodeljuje pripadnost opreme društvu in avtomatsko dodeljen primarni ključ pregleda (*pregled_id*).
- V entiteti **izposoja** so shranjeni podatki o izposojah opreme. Atributi entitete so: datum izposoje opreme (*izposojeno*), datum vrnitve opreme (*vrnjeno*), opombe o izposoji oziroma stanju opreme ob vrnitvi ali izposoji (*opombe*) in podatki o osebi, ki si je opremo izposodila (*izposodil*). Entiteta vsebuje še atribut *izposoja_id*, ki je primarni ključ entitete in se avtomatsko dodeli ob vsakem vnosu.

- Entiteta **izposoja_oprema** je namenjena dodelitvi izposoj (entiteta *izposoja*) posamezni opremi (entiteta *oprema*). Vsaka oprema ima lahko vnešenih več izposoj. Entiteta ima poleg identifikacijskih atributov opreme (*oprema_id*) in izposoje (*izposoja_id*) še primarni ključ (*izposoja_oprema_id*) ter opsijsko polje za vnos količine izposojene opreme (*kolicina*).

3.1.5 Intervencije

Sklop vsebuje pet entitet:

- Entiteta **intervencija** vsebuje primarne podatke o opravljeni intervenciji. Atributi entite so: čas prejetega poziva (*int_poziv*), točna lokacija intervencije (*lokacija*, *kraj*) in identifikator društva (*drustvo_id*). Ti atributi so obvezni. Zaradi možnosti ustvarjanja intervencije že med potekom le-te, ko nimamo še vseh podatkov, pa entiteta vsebuje še druge neobvezne attribute: čas konca intervencije (*cas_konec*), čas izvoza na intervencijo (*int_izvoz*), čas prihoda na lokacijo (*int_lokacija*), koordinati intervencije X in Y (*lokacijaX*, *lokacijaY*), podatek o porabljenih gasilnih sredstvih (*gasilna_sredstva*), opis (*opis*) in podatke o poškodovanih osebah (*poskodovani*), dodatno obveščenih službah (*dodatno_obvescen*) ter drugih prisotnih na intervenciji (*drugi_prisotni*). Atribut *vrsta_int_id* pa intervenciji dodeljuje vrsto intervencije iz entite **si_vrste_intervencij**.
- Entiteta **si_vrste_intervencij** je šifrant, ki vsebuje podatke o vrstah intervencij. Entiteta ima poleg primarnega ključa *vrsta_int_id* še atribut z opisom vrste intervencije (*vrsta_int*).
- Entiteta **intervencija_clani** hrani podatke o članih, ki so se udeležili posamezne intervencije. Vsak član se lahko udeleži več intervencij in ene intervencije se lahko udeleži več članov. Atributi entite sta identifikatorja intervencij (*intervencija_id*) in članov (*clan_id*) ter podatek, ali je bil član na intervenciji samo dežuren (*dezursto*).

- V entiteti **intervencija_oprema** sta poleg identifikacijskih atributov intervencije (*intervencija_id*) in opreme (*oprema_id*) še atributa o poškodovani opremi (*poskodovana*) in opombah pri uporabi opreme (*opomba*). Atribut *intervencija_oprema_id* je primarni ključ entitete in je avtomatsko dodan ob vsakem vnosu.
- Entiteta **intervencija_vozila** je namenjena dodelitvi vozil (entiteta *vozilo*), posamezni intervenciji (entiteta *intervencija*). Entiteta ima poleg identifikacijskih atributov vozila (*vozilo_id*) in intervencije (*intervencija_id*) še primarni ključ *intervencija_vozilo_id*.

3.1.6 Vaje

Sklop vsebuje tri entitete:

- Entiteta **vaje** vsebuje podatke o vajah v društvu. Atributi entitete so: čas pričetka vaje (*vaja_zacetek*), čas konca vaje (*vaja_konec*), naziv vaje (*vaja_naziv*), lokacija izvedbe (*lokacija*), opis vaje (*opis_vaje*) ter podatek o članu, ki je izpolnil poročilo (*porocila_izpolnil*). Atributa *tip_vaje_id* in *društvo_id* pa vaji dodeljujeta tip vaje in pripadnost društvu iz entitet **si_tipi_vaj** in **društvo**.
- Entiteta **si_tipi_vaj** je šifrant, ki vsebuje vrste vaj. Entiteta vsebuje atributa opis vrste vaje (*vrsta_vaje*) in primarni ključ (*tip_vaje_id*).
- Entiteta **vaja_clani** je namenjena dodelitvi vaj posameznemu članu. Entiteta ima poleg identifikacijskih atributov vaje (*vaja_id*) in člana (*clan_id*) še primarni ključ *vaja_clani_id*. Vsak član je lahko prisoten na več vajah.

3.1.7 Novice

Entiteta **novice** je samostojen sklop, ki je namenjen shranjevanju novic iz predstavitvene spletne strani. Entiteta vsebuje attribute: naslov novice (*na-*

slov), datum vpisa novice (*ustvarjena*), podatki o avtorju (*avtor*, *avtorImePriimek*), vsebina novice (*vsebina*), povezava do slike (*slika*), kategorija novice (*kategorija*) in ikona za prikaz (*ikona*). Poleg tega vsebuje še atribut *novica_id*, ki je primarni ključ in se avtomatsko doda vsakemu vnosu.

3.1.8 Uporabniki

Gre za ločen sklop treh entitet, ki se uporabljajo za hranjenje podatkov o registriranih uporabnikih na portalu in njihovih pravic za dostop do posameznih delov portala.

- Entiteta **AspNetUsers** vsebuje podatke o registriranih uporabnikih portala. Entiteta vsebuje attribute: naslov elektronske pošte (*Email*), uporabnikovo geslo (*PasswordHash*), uporabniško ime (*UserName*), podatek, ali je uporabnik potrdil naslov elektronske pošte (*EmailConfirmed*) in unikatni identifikator člana (*Id*).
- Entiteta **AspNetRoles** hrani podatke o vlogah na portalu. Vsebuje atributa ime vloge (*Name*) in unikatni identifikator vloge (*Id*), ki je avtomatsko dodeljen vsakemu vnosu.
- Entiteta **AspNetUserRoles** je namenjena dodelitvi vlog posameznemu uporabniku. Entiteta vsebuje le pare unikatnega identifikatorja uporabnika (*UserId*) in vloge (*RoleId*).

Poglavje 4

Komponente spletnega portala

Poleg izdelave entitetnega modela je izdelava spletnega portala obsegala še dva dela. Prvi del je bila izdelava uporabniškega vmesnika portala, drugi del pa sta predstavljali izdelava logike na strani strežnika in že prej omenjena izdelava modela. Zaradi lažjega dela sem se odločil za simultani razvoj vseh delov. Tako sem portal razvijal po sklopih, katere bom predstavil v nadaljevanju.

4.1 Spletni vmesnik

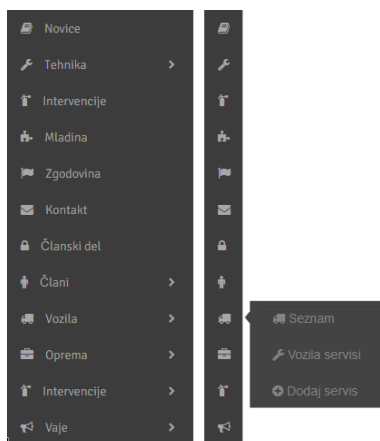
Spletni vmesnik je razvit na podlagi Bootstrap predloge UniconAdmin[18]. Zaradi uporabe ASP.NET ogrodja sem predlogo prilagodil, tako da je sestavljena iz treh delov. Na sliki 4.1 je prikazana razdelitev spletnega vmesnika.



Slika 4.1: Razdelitev spletnega vmesnika.

Razdelitev in postavitve spletnega vmesnika oziroma spletne je definirana v datoteki `_Layout.cshtml`. Deli spletne strani so:

1 - Stranski navigacijski meni: Stranski meni je namenjen navigaciji po spletnem portalu. Po prijavi uporabnika, ki ima dodeljene ustrezne vloge, se na seznamu poleg menija predstavitvene strani pojavijo še opcije za uporabniški portal. Slika 4.2 prikazuje prilagajanje navigacijskega menija glede na širino okna brskalnika.

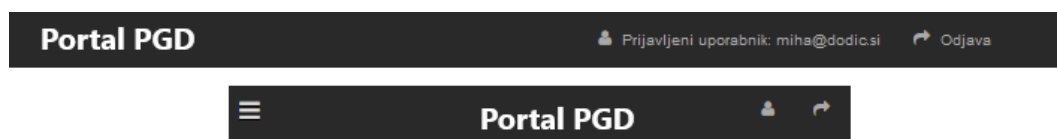


Slika 4.2: Navigacijski meni pri širokem oknu brskalnika (levo) in pri ozkem oknu brskalnika (desno), kjer je pri slednjem prikazan tudi izvlečni meni.

2 - Vrstica z uporabniškimi orodji: Vrstica služi za prijavo in odjavo uporabnika in kot bližnjica do strani za urejanje uporabnikovih nastavitev. Ko je uporabnik prijavljen, lahko s klikom na uporabniško ime oziroma ikono (slika 4.3), ki se nahaja v pasici, dostopa do svojih uporabniških nastavitev. Pasica je v ogrodje dodana kot delna stran, kar je prikazano v izvorni kodi 4.1.

Izvorna koda 4.1: Klic prijavnne vrstice.

```
<div id="user-nav">  
    @Html.Partial("_LoginPartial")  
</div>
```

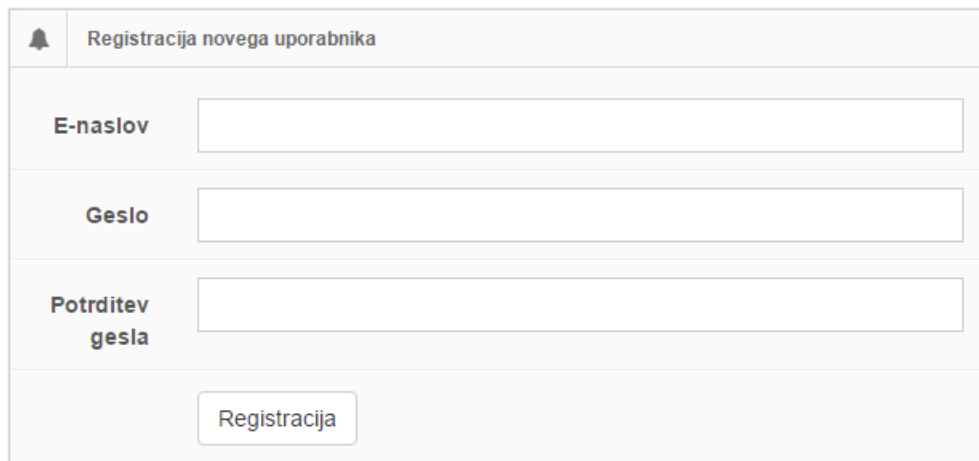


Slika 4.3: Primer uporabniške pasice pri različni širini brskalnika. Zgoraj je prikazana vrstica pri širokem oknu brskalnika, spodaj pa je prikazana vrstica pri ozkem oknu brskalnika.

3 - Polje za prikaz vsebine: Polje za prikaz vsebine je s strani *_Layout.cshtml* prikazano s pomočjo funkcije *RenderBody()*, ki je del osnovne knjižnice ASP.NET. V tem polju se prikazujejo podatki, ki jih nadzornik pridobi iz modela. Za potrebe spletnega portala je bilo ustvarjenih deset modelov in dva nadzornika. S strani, na kateri se nahaja uporabnik, s funkcijo *ViewBag* v ogrodje strani pošljamo naslov trenutne strani, ki uporabniku služi za lažjo navigacijo po portalu.

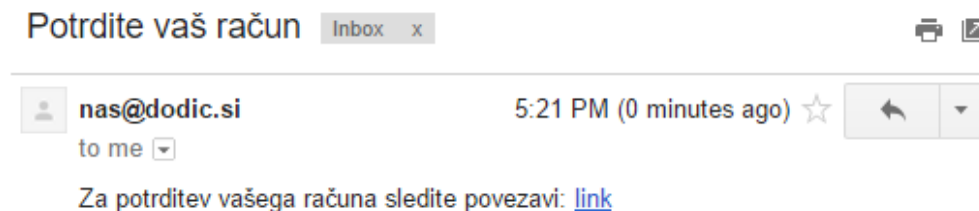
4.2 Registracija in prijava

Ob obisku spletne strani brez prijave imajo obiskovalci dostop samo do predstavitvene strani društva. V kolikor obiskovalci želijo dostopati do članskih vsebin, se morajo registrirati v portal (slika 4.4). Uporabniki za registracijo potrebujejo e-naslov in geslo. Izbrano geslo mora ustrazati naslednjim zahtevam: dolžina vsaj 8 znakov, vsebovati mora male črke, velike črke, številke in posebne znake. Sistem ob prijavi preverja tudi, ali je uporabnik vnesel veljavno obliko e-naslova. Po opravljeni registraciji uporabnik na svoj e-naslov prejme sporočilo s povezavo do potrditve računa (slika 4.5).



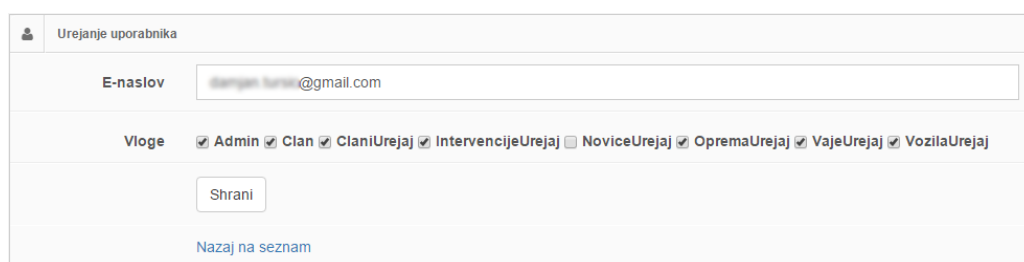
The image shows a registration form titled "Registracija novega uporabnika" (Registration of a new user). It contains three input fields: "E-naslov" (Email), "Geslo" (Password), and "Potrditev gesla" (Confirm password). Below the fields is a button labeled "Registracija".

Slika 4.4: Maska za registracijo novega uporabnika.



Slika 4.5: Sporočilo za potrditev računa.

Po opravljeni registraciji in potrditvi e-naslova uporabnik še vedno nima omogočenih dodatnih dostopov. Administrator portala mu mora preko konzole za upravljanje uporabnikov dodeliti ustrezne dostope. Šele ko je to narejeno, lahko uporabnik dostopa do članskih vsebin. Za tak način sem se odločil zaradi možnosti registracije uporabnikov, ki niso člani društva in niso upravičeni do vpogleda v podatke društva. Na sliki 4.6 vidimo administratorsko masko za urejanje dostopov uporabnikov.



Urejanje uporabnika	
E-naslov	<input type="text" value="djanjan.kurjak@gmail.com"/>
Vloge	<input checked="" type="checkbox"/> Admin <input checked="" type="checkbox"/> Clan <input checked="" type="checkbox"/> ClaniUreaj <input checked="" type="checkbox"/> IntervencijeUreaj <input type="checkbox"/> NoviceUreaj <input checked="" type="checkbox"/> OpremaUreaj <input checked="" type="checkbox"/> VajeUreaj <input checked="" type="checkbox"/> VozilaUreaj
<input type="button" value="Shrani"/>	
Nazaj na seznam	

Slika 4.6: Maska za urejanje uporabnikovih dostopov.

4.2.1 Uporabniške pravice

Zaradi omejevanja dostopov do posameznih strani je bilo ustvarjenih več uporabniških vlog. Na sliki 4.7 je prikazano, katere dostope omogoča posamezna vloga. Vsakemu članu društva je najprej dodeljena vloga *Član*. S to vlogo mu je odobren vpogled v vse vodene evidence društva. V kolikor uporabnik potrebuje dodatne dostope za urejanje posameznih sklopov, mu jih mora dodeliti administrator portala. Uporabniški dostopi na portalu se urejajo v nadzorniku, s pomočjo funkcije *Authorize*. V izvorni kodi 4.2 je prikazano, kako v nadzorniku določimo, katero vlogo morajo imeti uporabniki dodeljeno, da lahko dostopajo do te strani. V kolikor uporabnik nima dodeljene potrebne vloge, ga nadzornik preusmeri na stran za prijavo v sistem. Dostopi se urejajo tudi v pogledu, saj so posamezni gumbi vidni le uporabnikom z ustreznimi pravicami. Tako na primer gumb za dodajanje novic na osnovni strani vidijo samo uporabniki z dodeljeno vlogo *Novice Ureaj* (slika 4.8).

	Admin	Clan	Clani Urejaj	Intervencije Urejaj	Novice Urejaj	Oprema Urejaj	Vaje Urejaj	Vozila Urejaj
Urejanje uporabnikov in vlog								
Seznam članov								
Član podrobno								
Dodaj/uredi člana								
Seznam zdravniških pregledov								
Dodaj/uredi zdravniški pregled								
Seznam vozil								
Vozilo podrobno								
Seznam servisov vozila								
Dodaj/uredi servis vozila								
Seznam opreme								
Oprema podrobno								
Dodaj/uredi opremo								
Izposoja opreme								
Dodaj/uredi izposajo opreme								
Seznam intervencij								
Intervencija podobno								
Dodaj/uredi intervencijo								
Seznam vaj								
Vaja podrobno								
Dodaj/uredi vajo								
Dodaj/uredi novico glavna stran								
Dodaj/uredi novico mladina								

Slika 4.7: Seznam dostopov.

Izvorna koda 4.2: Omejevanje dostopov uporabnikov do posameznih strani.

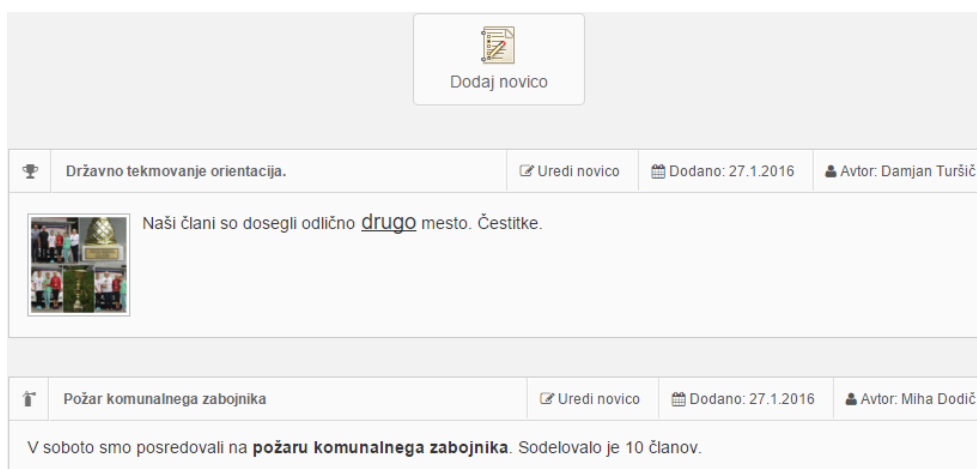
```
[Authorize(Roles = "ClaniUrejaj")]
public ActionResult ClaniDodaj(string drustvo){
    ...
    return View(model); }
```

4.3 Predstavitvena spletna stran

Za potrebe predstavitve gasilskega društva sem v okviru portala izdelal tudi predstavitveno spletno stran. Tu gre za statične vsebine, ki jih je ob spremembah potrebno urejati ročno preko urejevalnika HTML in se ne generirajo dinamično. Izjema so samo novice, ki se vnašajo preko spletnega obrazca in se shranjujejo v podatkovno bazo.

4.3.1 Novice

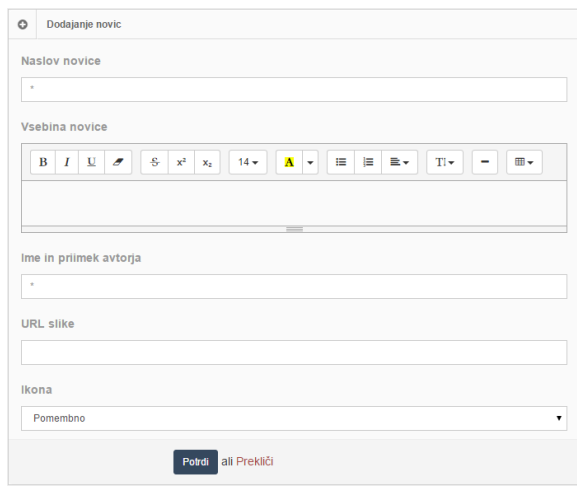
Uporabniki, ki imajo dodeljeno vlogo *Novice Urejaj*, imajo možnost vnosa in urejanja novic na glavni strani ter na strani, namenjeni prvenstveno mlajšim članom. Novice so shranjene v podatkovni bazi in so preko modela prikazane v ustreznem pogledu (slika 4.8).



Slika 4.8: Prikaz novic in gumb za dodajanje novic, ki je viden samo uporabnikom z dodeljeno vlogo *Novice Urejaj*.

Obrazec za vpis nove novice od uporabnika zahteva vnos naslova, vsebine, avtorja, povezave do slike in izbiro ikone, ki bo prikazana ob novici (slika 4.9). Polja, ki jih mora uporabnik obvezno vnesti, so označena z znakom *. Polje *Vsebina novice* uporablja WYSIWYG urejevalnik Summernote. Vtičnik s

pomočjo knjižnice jQuery in JavaScripta spremeni klasično vnosno polje v polje z možnostjo urejanja besedila (izvorna koda 4.3). Tako lahko uporabniki brez uporabe značk HTML spreminjajo velikost, barvo in druge lastnosti besedila, vstavljajo tabele, izbirajo poravnavo besedila in spreminjajo druge lastnosti, ki jih omogočajo urejevalniki besedil in ki jih dovoljuje HTML.



Slika 4.9: Forma za vnos nove novice.

Izvorna koda 4.3: JavaScript za inicializacijo urejevalnika besedila.

```
(function ($) {
    function HomeIndex() {
        var $this = this;
        function initialize() {
            $('#Vsebina').summernote({
                focus: true, height: 200,
                codemirror: {theme: 'united'},
                toolbar:
                [ ['style', ['bold', 'italic', 'underline', 'clear']],
                  ['font', ['strikethrough', 'superscript', 'subscript']],
                  ['fontsize', ['fontsize']],
                  ['color', ['color']],
                  ['para', ['ul', 'ol', 'paragraph']]
                ],
            });
        }
    }
})
```



```
        ['height', ['height']],  
        ['insert', ['hr']],  
        ['table', ['table']] ]  
    }); }  
    $this.init = function () {  
        initialize();  
    } }  
    $(function () {  
        var self = new HomeIndex();  
        self.init(); })  
    }(jQuery))
```

4.3.2 Galerija

Na predlog testnih uporabnikov sem na predstavitveni spletni strani implementiral tudi galerijo slik. Galerija se uporablja na straneh, namenjenih za predstavitev vozil in opreme društva (slika 4.10). Gre za skripto jQuery, ki seznam slik pretvori v interaktivno galerijo.



Slika 4.10: Primer uporabe galerije slik na strani, namenjeni predstavitvi posameznega vozila.

4.4 Del za registrirane uporabnike

Glavni namen spletnega portala je sistem za vodenje evidenc društva. Portal je dostopen samo registriranim uporabnikom z dodeljenimi ustreznimi vlogami. Uporabniki, ki jim je dodeljena vloga *Član* imajo dostop do branja vseh evidenc. V kolikor želijo uporabniki vnašati podatke za posamezen sklop, jim mora biti odobrena dodatna vloga. V nadaljevanju bom predstavil posamezne sklope in njihove posebnosti.

4.4.1 Člani

V sklopu **Člani** so uporabnikom dostopni vsi podatki o članih. Tu lahko vnašamo in urejamo člane, pregledujemo stanje zdravniških pregledov, vnašamo nove zdravniške preglede in si ogledamo podrobnosti posameznega člana.


Zdravniški pregledi

Uporabniki lahko izbirajo med tremi različnimi pogledi za zdravniške preglede:

- prikaz vseh zdravniški pregledov ne glede na njihov status (prikažejo se tudi zdravniški pregledi, ki jim je potekel datum veljavnosti),
- prikaz samo veljavnih zdravniških pregledov,
- prikaz članov brez opravljenih zdravniških pregledov (prikažejo se nam samo operativni člani in operativni člani rezerve brez opravljenih zdravniških pregledov, ne pokažejo pa se člani, ki nimajo opravljenega tečaja za gasilce).

V polje *Veljavnost ZP* je uporabniku zaradi hitrejšega vpogleda z uporabo barvnih oznak prikazano stanje zdravniškega pregleda (slika 4.11). Rdeča oznaka pomeni, da je veljavnost zdravniškega pregleda potekla, zelena pomeni da je zdravniški pregled še vedno veljaven, modra pa, da bo veljavnost

zdravniškega pregleda potekla v manj kot mesecu dni. Ker je pogoj za opravljanje zdravniškega pregleda opravljen tečaj za gasilca, se lahko zdravniške preglede dodaja samo članom z dodeljeno vrsto člana operativni gasilec ali operativni gasilec rezerva.

Zdravniški pregledi člana				Pokaži člane brez ZP	Pokaži samo veljavne	Dodaj zdravniški
	Ime in priimek	Vrsta ZP	Datum opravljanja	Datum poteka	Številka potrdila	Veljavnost ZP
	Miha Dodič	4. člen	01.07.2016	01.05.2016	testxxx	Veljaven
	Aleš Debevec	4. člen	19.11.2015	19.02.2016	Potrdilo 22	POZOR
	Jože Trček	4. člen	12.12.2014	12.12.2017	ZD Ljubljana Metelkova	Veljaven
	Andrej Jereb	4. člen	18.10.2011	18.10.2014	ZD Vrhnika	Potekel
	Miha Dodič	4. člen	11.08.2008	11.08.2011	73971/08.08	Potekel

Slika 4.11: Pregled zdravniških pregledov ne glede na status in barvne oznake s prikazom stanja zdravniškega pregleda.

Podrobni pregled člana

V tem pogledu so zbrani vsi podatki o članu. Uporabniku so vidni osnovni podatki o članu, opravljenih tečajih, udeležbi na vajah, udeležbi na intervencijah in podatki o zdravniških pregledih (slika 4.12).

Polja za prikaz opravljenih tečajev, prisotnosti na intervencijah, prisotnosti na vajah in opravljenih zdravniških pregledih so prikazana kot svoja delna stran. Izvorna koda 4.4 prikazuje dodajanje podatkov o zdravniških pregledih na stran člana, če ima član opravljen vsaj en zdravniški pregled oziroma izpis, da član nima opravljenih zdravniških pregledov. Pri izpisu udeležbe na intervencijah in vajah se uporabniku prikaže tudi skupno število opravljenih delovnih ur člana pri posameznem sklopu (intervencije ali vaje).

Član podrobnosti

Ime: Miha
Priimek: Dodič
Naslov: Zabočevo 4, 1353 Borovnica
Kontakt: 031618865 miha@dodic.si

Opravljeni tečaji:
Tečaj za tehničnega reševalca
Uspodabljanje za delo z helikopterjem
Tečaj za nosilca izolirnega dihalnega aparata
Tečaj za gašenje notranjih požarov B
Izpit za C kategorijo

Udeležba na intervencijah:

Vrsta intervencije	Čas poziva	Čas konca	Lokacija	Kraj	Opis intervencije	Trajanje	
Požar na prometnih sredstvih	2.8.2014 - 22:00	3.8.2014 - 3:00	Paplerjeva ulica 6	Borovnica	Požar stanovanjskega objekta. Izvajanje notranjih napadov z IDA, varovanje sosednjih objektov.	05:00:00	i
Požar na prometnih sredstvih	10.3.2015 - 10:14	10.3.2015 - 12:15	Ohonica 4	Ohonica	Odpiranj vrata, ter pomoč NMP pri prenosu obolele osebe.	02:01:00	i

Skupaj opravljenih ur na intervencijah: 7,01

Udeležba na vajah:

Naziv vaje	Opis	Lokacija	Vrsta vaje	Začetek	Konec	Trajanje	
Mesečne vaje marec	Pregled osebne zaščitne opreme. Vtek MB Rosenbauer ter agregata.	Gasilski dom Brezovica pri Borovnici	Redne mesečne vaje	10.3.2013 - 19:00	10.3.2013 - 20:15	01:15:00	i
Vaja test	Vaja test 2	Gasilski dom	Vsposabljanje	1.2.2000 - 19:00	1.2.2000 - 20:00	01:00:00	i

Skupaj opravljenih ur na vajah: 2,25

Zdravniški pregledi:

Vrsta ZP	Datum opravljanja	Datum poteka	Številka potrdila	Veljavnost ZP
4. člen	01.07.2016	01.05.2016	testbox	Veljaven
4. člen	15.10.2014	15.10.2017	ZD Ljubljana	Veljaven
3. člen	15.10.2014	15.10.2019	ZD Ljubljana	Veljaven
4. člen	11.08.2008	11.08.2011	73971/08.08	Potekel

Slika 4.12: Podrobni pregled izbranega člana.

Izvorna koda 4.4: Izpis delne strani o zdravniških pregledih člana oziroma izpis, da član nima opravljenih zdravniških pregledov.

```
if (portalD.Models.ZdravniskiModel.GetAllZdravniskiClana(Model.
    ClanId).Count() != 0){
    @Html.Partial("~/Views/Home/_ClanZdravniskiPartial.cshtml");
}else{
    <p>Clan nima opravljenega zdravniškega pregleda.</p>}
```

4.4.2 Oprema

V sklopu **Oprema** se vodijo podatki o opremi. Uprabniki lahko dostopajo do seznama opreme, dodajajo in urejajo posamezno opremo ter dodajajo podatke o izposoji in pregledih opreme.

Izposoja opreme

Evidenc izposoje opreme se v društvu ni vodilo. Zaradi tega se je že dogajalo, da so člani iskali opremo, ki si jo je izposodil kateri izmed članov. Da v prihodnje ne bi prihajalo do tega, sem implementiral modul, ki nam omogoča vodenje evidence o izposoji opreme. Tu člani vnašajo podatke o izposoji posamezne opreme. Podatki o posamezni izposoji opreme se hranijo v dveh entitetah podatkovne baze. Zaradi tega sem pri vnosu nove izposoje v podatkovno bazo uporabil ukaz SQL *SELECT SCOPE_IDENTITY()*, ki nam vrne identifikator zadnjega vnosa v podatkovno bazo (izvorna koda 4.5). Ta identifikator je potreben za vnos podatkov v drugo entiteto.






Izvorna koda 4.5: SQL in C# koda za vpis nove izposoje v podatkovno bazo.

```
using (SqlCommand com = new SqlCommand(@"INSERT INTO [dbo].[
    izposoja]
([izposojeno], [vrnjeno], [opombe], [izposodil])
VALUES
(@izp, @vrnjeno, @opombe, @izposodil)
SELECT SCOPE_IDENTITY() AS id")){
```

```
command.Parameters.AddWithValue("@izp", this.IzposojaDatum);
command.Parameters.AddWithValue("@vrnjeno", this.VrnitevDatum);
command.Parameters.AddWithValue("@opombe", (object)this.Opombe
    ?? DBNull.Value);
command.Parameters.AddWithValue("@izposodil", this.Izposodil);
DataTable table = database.GetDataTable(command);
if (table.Rows.Count == 1){
    id = table.Rows[0].GetColumnValue("id", 0);}
}
using (SqlCommand command1 = new SqlCommand(@"INSERT INTO [dbo]
    [izposoja_oprema]
    ([oprema_id], [izposoja_id], [kolicina])
    VALUES
    (@opremaId, @izposojaId, @kolicina)")){
command1.Parameters.AddWithValue("@opremaId", this.OpremaId);
command1.Parameters.AddWithValue("@izposojaId", id);
command1.Parameters.AddWithValue("@kolicina", this.Kolicina);
database.ExecuteSimpleCommand(command1);
}
```

Pregledi opreme

Pregledi oziroma servisi opreme se lahko vnašajo samo za opremo, ki ima označeno izbiro, da se zanjo opravljajo periodni pregledi. Ob vnosu pregleda mora uporabnik vnesti tudi datum veljavnosti pregleda. Pregled opreme, za katero se opravljajo servisni oziroma kontrolni pregledi, ima namreč omejen rok veljavnosti. Po pretečenem roku je potrebno ponovno opraviti pregled. Slika 4.13 prikazuje seznam opreme skupaj z akcijskimi gumbi, ki so na voljo uporabnikom z dodeljeno vlogo *Oprema Ureja*. V prvem stolpcu je tako gumb za urejanje podatkov o opremi, v predzadnjem gumb za dodajanje pregleda opreme in v zadnjem gumb za dostop do podrobnosti opreme. Pri drugem vnosu je vidno, da za opremo, ki nima označenega opravljanja periodnega pregleda, ni mogoče vnesti pregleda opreme.

	Naziv	Leto proizvodnje	Opis	Serijska št.	Količina		
	IDA Drager	2000	IDA Drager - st 1	DR22323	1		
	Miza	1985			20		

Slika 4.13: Pregled posamezne opreme z akcijskimi gumbi. V prvem stolpcu je gumb za urejanje opreme, v predzadnjem stolpcu je gumb za dodajanje pregleda in v zadnjem stolpcu gumb za dostop do podrobnosti opreme.

4.4.3 Vozila

Sklop **Vozila** služi za vodenje podatkov o vozilih in njihovih servisnih pregledih. Zaradi redkega dodajanja novih vozil nisem implementiral forme za vnos novega vozila. Tako je potrebno vsako novo vozilo ročno vnesti v podatkovno bazo. Prav tako je potrebno ročno vnašati tipe vozil glede na tipizacijo GZS, ki pa so za trenutno veljavno tipizacijo GZS že vneseni. Slika 4.14 prikazuje vnos novega servisnega pregleda s poljem za izbiro vozila, za katerega se vpiše nov pregled. Polje s ponujenimi vozili se avtomatsko napolni s seznamom vozil, ki so vnesena v podatkovno bazo, preko funkcije *GetAllVozilaDropdown*. Izvorna koda 4.6 prikazuje polnjenje spustnega seznama z vozili iz podatkovne baze.

+

Dodaj servis vozila

Identifikator vozila

GVV-1 VW LT45

GVV-1 VW LT45

GVM-1 Peugeot Exprt

04/01/2016

*

Opis servisa

Potrdi

ali Prekliči

Slika 4.14: Forma za vnos novega servisnega pregleda za posamezno vozilo.

Izvorna koda 4.6: Polnjenje spustnega seznama.

```
@Html.DropDownListFor(m => m.VoziloId, portalD.Models.  
    VozilaModel.GetAllVozilaDropdown(drustvoId), new { @class  
    = "form-control input-sm" } )
```

4.4.4 Intervencije

Sklop **Intervencije** služi za vodenje evidenc o opravljenih intervencijah in prisotnosti članov na posamezni intervenciji. Tu lahko uporabniki dodajajo in urejajo intervencije ter pregledujejo podatke o preteklih intervencijah.

Dodajanje intervencij

Uporabniki lahko novo intervencijo dodajo že med samim potekom intervencije. Obvezna polja za vnos nove intervencije so: vrsta, čas poziva, čas izvoza, lokacija in opis intervencije. Ostala polja lahko uporabnik izpolni kasneje. Vsa datumska polja so izvedena s pomočjo Bootstrap vtičnika *DateTimePicker*. Ta uporabnikom omogoča enostaven vnos datuma in ure. Na sliki 4.15 vidimo formo za vnos nove intervencije z vtičnikom za izbiro datuma in ure ter seznam izbirnih polj za člane. Med ustvarjanjem intervencije lahko uporabnik izbere tudi prisotne člane na intervenciji. Za dodajanje članov sem uporabil seznam izbirnih polj[17], ki prikaže vse operativne člane in operativne člane rezerve. Pri urejanju intervencije so pri članih, ki so bili izbrani za prisotnost na intervenciji, polja označena, pri ostalih pa ne. Izvorna koda 4.7 prikazuje generiranje novega seznama izbirnih polj v nadzorniku.

Izvorna koda 4.7: Generiranje seznama izbirnih polj s člani za vnos prisotnosti na intervenciji.

```
var vsiClani = portalD.Models.ClanModel.GetClaneDrustva(  
    drustvoModel.DrustvoId);  
var checkBoxListItems = new List<CheckBoxListItem>();  
foreach (var clan in vsiClani){  
    checkBoxListItems.Add(new CheckBoxListItem(){
```



```
        ID = clan.ClanId,  
        Display = clan.ImePriimek,  
        IsChecked = false })  
    ;}  
model.ClaniIntervencija = checkBoxListItems;
```

Dodaj intervencijo

Vrsta intervencije
Požar na prometnih sredstvih

Čas poziva
Polje je obvezno

Čas izvoza
29. 01. 2016 20:05

Čas prihoda na lokacijo
29. 01. 2016 20:04

Lokacija intervencije
*

Kraj intervencije

Opis intervencije
*

Porabljena gasilna sredstva

Dodatno obveščene službe

Druge pristone službe

Poškodovani

Prisotni člani na intervenciji

- ☐ Miha Dodič
- ☐ Damjan Turšič
- ☐ Ciril Trček









Slika 4.15: Vnos nove intervencije s prikazanim vtičnikom za izbiro datuma in ure.

4.4.5 Vaje

V sklopu **Vaje** se vodijo podatki o vajah. Uporabniki lahko pregledujejo evidence o preteklih vajah in dodajajo ter urejajo posamezne vaje. Na voljo je pregled vaj glede na izbrano leto vaje ali podroben pregled posamezne vaje.

4.4.6 Seznam vaj

Uporabniki lahko dostopajo do seznama vaj. Ob izbiri seznama vaj iz stranskega navigacijskega menija se uporabniku prikažejo vse opravljene vaje. Kasneje lahko uporabnik prilagaja pogled in izbira leto za prikaz vaj. Na sliki 4.16 vidimo osnovni prikaz vseh vaj.

Vaje			Vse vaje	2000	2003	2004	2013	2014	2015
	Naziv vaje	Vrsta vaje	Začetek	Konec	Trajanje	St. članov	Trajanje skupaj		
	Mesečne vaje marec	Redne mesečne vaje	10.3.2013 - 19:00	10.3.2013 - 20:15	01:15:00	3	03:45:00		
	Prva pomoč april	Prva pomoč	10.4.2013 - 19:00	10.4.2013 - 20:00	01:00:00	3	03:00:00		
	Tečaj IDA	Tečaj	10.1.2014 - 8:00	10.1.2014 - 16:00	08:00:00	1	08:00:00		
	test test	Tečaj	1.12.2015 - 20:00	1.12.2015 - 21:00	01:00:00	2	02:00:00		

Slika 4.16: Prikaz seznama vseh vaj.

Poglavje 5

Sklepne ugotovitve

V okviru diplomskega dela sem izdelal delujoč spletni portal za predstavitev gasilskega društva in vodenje društvenih evidenc. Do sedaj je bil sistem dostopen le omejenemu številu uporabnikov, zato ga bo potrebno pred dejansko objavo testirati z večjim številom uporabnikov. Med razvojem spletnega portala se je izkazalo, da bi bil sistem dobrodošel za uporabo tudi pri drugih gasilskih društvih. Nekatere funkcionalnosti, ki so bile razvite že pred tem, sem že uspel prilagoditi, nekaj pa jih bo še potrebno prilagoditi.

S trenutno verzijo spletne aplikacije sem zadostil vsem ciljem, ki sem si jih zadal pri izdelavi diplomskega dela. Uspešno sem implementiral evidenčne obrazce, ki so se vodili ročno, v elektronsko obliko. Dodane so še dodatne evidence, ki se pred tem niso vodile, kot so na primer evidence o opravljenih zdravniških pregledih. Prav tako je bila uspešno izdelana predstavitevna spletna stran društva. Končni izdelek diplomskega dela je tako popolnoma delujoč spletni portal, ki je pripravljen za takojšnjo uporabo. Testiranje aplikacije je pokazalo, da bo potrebno nekatere evidence v gasilskem društvu še vedno voditi ročno. Težavo namreč predstavlja generacijski razpon članov društva. Starejši člani, ki še vedno opravljajo dela v društvu, nimajo pametnih mobilnih naprav oziroma niso večji dela z njimi, prav tako pa niso večji dela z računalnikom. Tako bo potrebno ob vpeljavi sistema v dejansko rabo poskrbeti tudi za takšne težave, oziroma bo potrebno izvesti vsaj kakšen

tečaj dela s spletnim portalom za vse, ki bi to potrebovali.

5.1 Odziv testnih uporabnikov portala

V času razvoja spletnega portala sem nekaterim članom društva omogočil dostop do portala. Prosil sem jih, da portal testirajo in mi podajo svoja mnenja oziroma opazke. Uporabniki so portal večinoma testirali še v zgodnji fazi razvoja, zato se je večina pripomb nanašala na funkcionalnosti oziroma napake, ki so bile v kasnejši fazi odpravljene. Ena izmed takih pomanjkljivosti je tudi neobstoječa gumba za preklic akcije pri dialogu za registracijo uporabnika, prikazanem na sliki 4.4, kar bo še moralo biti dodano. Vsi testni uporabniki so portal sprejeli pozitivno, so pa izpostavili, da bo za smiselno uporabo portala potrebno vse člane društva prepričati v uporabo portala. Le tako bodo podatki popolni in bo vpeljava portala smiselna. Na podlagi priporočil testnih uporabnikov sem na primer v portal dodal galerijo pri predstavitveni spletni strani, spremenil mesto vnosa novic za podstran namenjeno mladini in opravil še nekaj manjših grafičnih popravkov.

5.2 Nadaljnje delo

Kljub delujoči različici aplikacije se je med razvojem pokazalo še nekaj funkcionalnosti, ki jih bom lahko razvil. Poleg odpravljanja obstoječih napak in pomankljivosti imam tako še dovolj nalog za nadaljnji razvoj:

- prilagoditev portala za uporabo za več gasilskih društev,
- implementacija modula tekmovanja, kjer se bodo vodili podatki o tekmovanjih, ki so se jih udeležili člani društva in podatki o doseženih rezultatih,
- implementacija vodenja sledi sprememb v podatkovni bazi,
- implementacija vnosa in prikaza lokacije pri intervencijah (podatkovni model že vsebuje polja za vnos teh podatkov),

- implementacija obveščanja administratorjev o novih registriranih članih,
- implementacija maske za vnos novega vozila,
- implementacija brisanja posameznih vnosov.

Literatura

- [1] S. Malnarič (2013) *Kongresi gasilske zveze Slovenije* [Online]. Dosegljivo:
http://www.gasilec.net/uploads/datoteke/Nevenka/zgodovina_kongresi_gzs_2013.pdf.
[Dostopano 21. 1. 2016].
- [2] Vulkan. [Online]. Dosegljivo:
<https://apl.gasilec.net/Vulkan/>. [Dostopano 21. 1. 2016].
- [3] SPIN grafični prikaz. [Online]. Dosegljivo:
<http://spin.sos112.si/SPIN2/Javno/GIS/GraficniPrikaz.aspx>. [Dostopano 21. 1. 2016].
- [4] Microsoft Visual Studio. [Online]. Dosegljivo:
<https://www.visualstudio.com/>. [Dostopano 21. 1. 2016].
- [5] Upravljaliec paketov - NuGet. [Online]. Dosegljivo:
<https://www.nuget.org/>. [Dostopano 21. 1. 2016].
- [6] Seznam CLI jezikov. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/List_of_CLI_languages/. [Dostopano 21. 1. 2016].
- [7] Team Foundation Version Control. [Online]. Dosegljivo:
<https://msdn.microsoft.com/en-us/library/ms181237.aspx/>. [Dostopano 21. 1. 2016].

-
- [8] Programski jezik C#. [Online]. Dosegljivo:
[https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)/](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)). [Dostopano 21. 1. 2016].
- [9] Model pogled nadzornik. [Online]. Dosegljivo:
[https://msdn.microsoft.com/en-us/library/dd381412\(v=vs.108\).aspx/](https://msdn.microsoft.com/en-us/library/dd381412(v=vs.108).aspx/). [Dostopano 21. 1. 2016].
- [10] Jezik za označevanje nadbesedila. [Online]. Dosegljivo:
<https://sl.wikipedia.org/wiki/HTML/>. [Dostopano 21. 1. 2016].
- [11] Kaskadne stilske predloge. [Online]. Dosegljivo:
https://en.wikipedia.org/wiki/Cascading_Style_Sheets/. [Dostopano 21. 1. 2016].
- [12] Java Script. [Online]. Dosegljivo:
<https://sl.wikipedia.org/wiki/JavaScript/>. [Dostopano 21. 1. 2016]. <https://sl.wikipedia.org/wiki/JavaScript>
- [13] Bootstrap. [Online]. Dosegljivo:
<http://getbootstrap.com/>. [Dostopano 21. 1. 2016].
- [14] Galerija prettyPhoto. [Online]. Dosegljivo:
<http://www.no-margin-for-errors.com/projects/prettyphoto-jquery-lightbox-clone/>. [Dostopano 21. 1. 2016].
- [15] Summernote editor. [Online]. Dosegljivo:
<http://summernote.org/>. [Dostopano 21. 1. 2016].
- [16] SAP Power Designer. [Online]. Dosegljivo:
<http://powerdesigner.de/en/overview/>. [Dostopano 21. 1. 2016].
- [17] Vzpostavitev seznama izbirnih polj. [Online]. Dosegljivo:
<http://www.exceptionnotfound.net/simple-checkboxlist-in-asp-net-mvc/>. [Dostopano 21. 1. 2016].

-
- [18] Predloga Unicorn admin. [Online]. Dosegljivo:
<https://wrapbootstrap.com/theme/unicorn-admin-template-WB0F35928/>. [Dostopano 21. 1. 2016].